# Concept Paper for the

# Standard/Rapid Terrain Database Generation Capability (STDGC) in support of the Synthetic Environment (SE) Core Initiative


Prepared by:

US Army Program Executive Office Simulation, Training,

and Instrumentation Command (PEO STRI)

12350 Research Parkway

Orlando, Florida 32826-3276

# TABLE OF CONTENTS

# TABLE OF FIGURES

# 1. Overview

This Government document serves as a Concept Paper for the requirements, architecture, capabilities, and goals of the SE Core Standard/Rapid Terrain Database Generation Capability (STDGC). The STDGC is a critical part of the overall SE Core Initiative. To provide a context for the STDGC this paper will also describe the fundamental goals of the SE Core Initiative. Section 2 contains an overview of the Synthetic Environment (SE) Core concept and its requirements, while Section 3 provides an overview of the Database Generation capability within SE Core. The key principles for SE Core database generation are captured in Section 4. A set of architectural or "design" concepts have also been captured as described in Section 5. Section 6 describes key dependencies or assumptions. A top level procurement plan is provided in Section 8.

# 2. Synthetic Environment Core Overview

The Synthetic Environment (SE) Core is the virtual component of the integrated Live/Virtual/Constructive (LVC) Training Environment (TE). SE Core's mission is to provide a common virtual environment (CVE) integrated within the LVC TE, enabled by the LVC Integrated Architecture (IA). SE Core will develop the CVE by linking virtual simulations into a fully integrated training capability through a new Virtual Simulation Architecture under development within PM CATT. The CVE, as part of the LVC-IA, is used to link virtual systems to live, constructive, and joint, interagency, intergovernmental, and multinational (JIIM) systems.

SE Core components will be developed incrementally. SE Core will focus on the development of the CVE for virtual simulations. Initial requirements provide the capability to integrate into the LVC TE, Command, Control, Communications, Computers, Intelligence Surveillance and Reconnaissance (C4ISR) and JIIM through the LVC-IA (see Figure 1). Incrementally, SE Core will focus on development of interoperability with Future Combat System (FCS)/Future Force embedded virtual training systems, NRFTT and PTT.

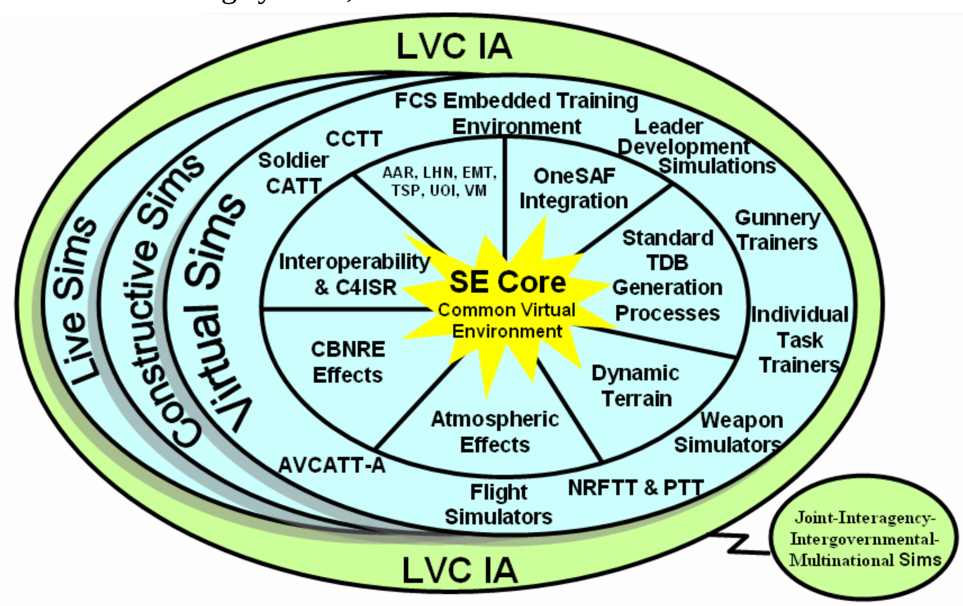


Figure 1: LVC-IA and SE Core Requirements

This concept paper identifies the high level requirements for the SE Core Standard/Rapid Terrain Generation Process and describes the preliminary design concepts for implementation of the solution. Other SE Core requirements that may impact the database generation capability, such as Dynamic Terrain requirements, are not directly addressed in this document but will be addressed in an incremental update to requirements after the initial award.

The Standard TDB Generation Process will itself be developed incrementally. CCTT, AVCATT-A and OOS will be some of the first programs SE Core will support. Other PEO-STRI programs will be added incrementally, along with the additional functionality required to support those programs. Long-term growth will include updates required to integrate related SE Core requirements, technology insertion and software maintenance. It is anticipated that the STDGC will evolve as new components and programs are brought into the SE Core fold.

### 2.1 SE Core Standard/Rapid Terrain Database Generation Requirements

The requirements for the SE Core Standard/Rapid Terrain Database Generation Capability (STDGC) include the generation of a 180 km x 180 km database at a resolution of data equivalent to NGA DTED level 3, an urban inset within that database that is 2.5 km x 2.5 km that has a resolution of data equivalent to NGA DTED level 5 and MOUT/ Urban operations capabilities. This database must be produced within 96 hours using automated processes. COTS tools must be provided that produce databases that are non-proprietary, open format, image generator (IG) independent, and consumable by multiple virtual simulations.

The eventual goal for STDGC is to be able to produce a 1000 km x 1000 km database at DTED level 1 with a 300 km x 300 km DTED level 3 inset and a 50 km x 50 km urban inset within 72 hours.

To achieve these goals, SE Core assumes that other initiatives within the Army or the DoD will address the quality of source data required to achieve the production timeline requirements of SE Core. Until that time the STDGC will also be required to allow manual manipulation of data and value adding through semi-automated and manual processes.

## 3. SE Core Database Generation Overview

A critical component within the overall SE Core concept is the ability to rapidly create terrain databases through centralized database generation facilities. This document captures a top-level view of concepts, objectives, and requirements based on existing and future programs for SE Core database generation capability. This capability is referred to as the Standard/Rapid Terrain Database Generation Capability (STDGC).

The overall SE Core concept covers multiple areas that are tangentially related to database generation or representations. This document only covers terrain database generation, spanning from available source data to the infrastructure required to generate run-time databases. Figure 2 decomposes terrain database generation into three components. The left-most section involves source data collection and refinement, which belong in the domain of organizations such as the National Geospatial-Intelligence Agency (NGA). The source data integration, terrain DB Generation, MDB, tools, APIs, etc. components are the primary focus of the STDGC. The right-hand section captures the concept of specialized run-time database formats used in specific applications or programs. The STDGC accepts source data as input and generates a wide range of runtime database formats as output. All components wholly within the STDGC Scope box are maintained and used by SE Core personnel.

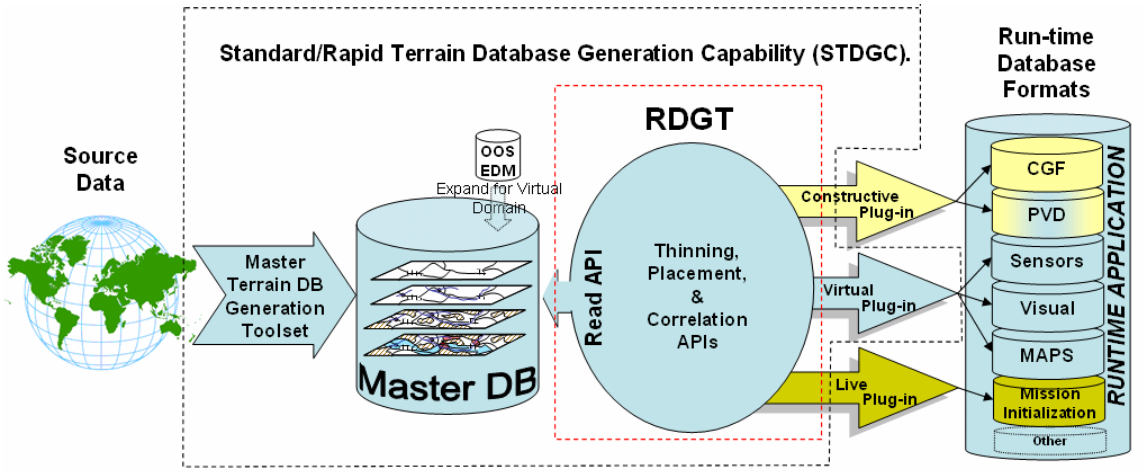


Figure 2: STDGC Overview

Organizationally, SE Core consists of a common, central set of capabilities, with confederate programs that use those capabilities. For database generation, SE Core will take on the full burden of executing and maintaining a database generation process, thereby relieving confederate programs of this concern. In the above diagram, the relationship between SE Core and confederate programs is illustrated by the fact that the APIs or “export plug-ins” are partly inside the STDGC scope and partly outside. The plug-ins are generally created (developed) by confederate programs, vendors, etc., while SE Core is responsible for specifying the plug-in architecture/interfaces (i.e. interface control document) and using plug-ins to export the run-time databases. Where the processing resources for new systems/programs support it, the run-time database generation will be done in real time. However, legacy systems will require that the run-time databases be built off line.

SE Core’s STDGC is based on a series of principles intended to avoid recurring issues historically encountered with database generation activities. Examples of these issues include development of highly-specialized toolsets and processes for different programs, heavy dependence on proprietary tools or individual contractors, and many different programs building the same area/database with only minor variations in content.

# 4. Key Principles

## 4.1 Cross-Program Commonality

A fundamental guiding principle of SE Core is *commonality*, both within virtual applications (via the CVE) and within the larger LVC-IA framework. To date, most larger simulation programs have devised their own specialized database generation capability. Where some reuse has occurred, such as the use of Commercial Off The Shelf (COTS) products for generation of OneSAF Testbed Baseline (OTB)/ Joint Semi-Automated Forces (JSAF) databases, the processes, sources, and end products still differ significantly. The STDGC is intended to address this by providing commonality in database generation from source data consumption through to an API for extraction of data.  Thus, each existing or new program can reuse the basics of database generation without starting from scratch.

A key component of cross-program commonality is the collaboration that must occur between SE Core and the various confederate programs.  SE Core is responsible for building databases for confederate programs.  This includes all database creating effort from source data to creation of run-time databases.  In return, confederate programs must provide detailed data requirements, iterate on these requirements with SE Core to determine good tradeoffs, then integrate with SE Core.  The detailed requirements include, as an example, an Environmental Data Model (EDM) for content, identification, and attribution.  As for tradeoffs, SE Core, and confederate programs, must meet "halfway."  In practice, this means that individual programs may be required to eliminate highly-specialized or complex data requirements.  This will serve to keep the STDGC capability more generic and cost-effective, while also improving run-time interoperability by encouraging commonality.  Integration efforts will involve test of databases produced by SE Core in a system context to ensure all requirements have been met.

## 4.2 Centralized

The STDGC is a reusable *capability* (which includes reusable *data*).  Rather than each program modifying, extending, and using the STDGC on their own, they reuse an overarching capability that is centrally maintained.  The STDGC will be created and maintained at a set of defined database generation facilities.  These specialized sites are then intended to support a wide range of end users/programs.  An Orlando facility would serve as the main site, while the various satellite sites would provide local capability to various organizations.  These sites would be connected by the Global Information Grid (GIG) and would share the same tools, data updates, and quality controls and configuration management processes.  While processes for configuration management and quality control will be common, each individual site may specialize in particular areas, such as detailed visual modeling of urban areas.  This concept is notionally captured in Figure 3:

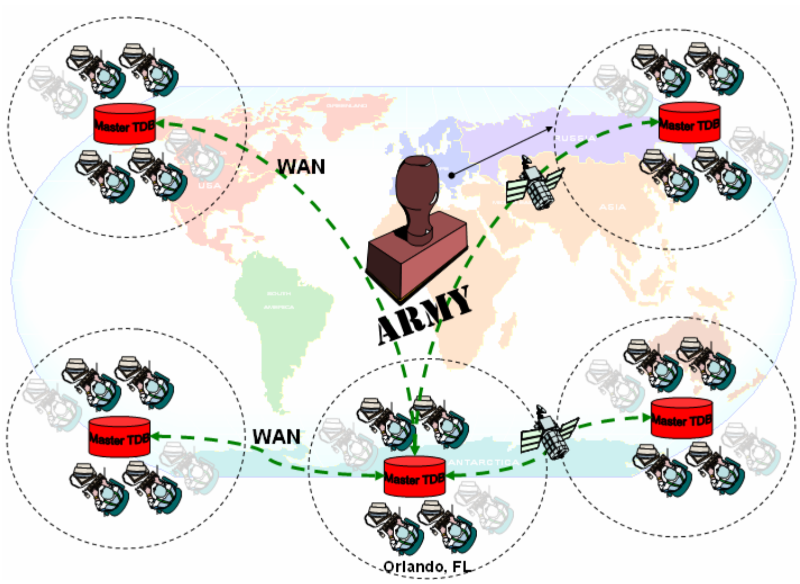


Figure 3: MDB Generation Centers (Global Information Grid)

## 4.3 Reuse of Datasets

The goal of broad reuse is supported by commonality and centralization. In this case, reuse is intended to cover more than just reuse of the STDGC itself. Reuse is expanded to include expertise via the personnel at centralized sites. Reuse of datasets becomes much easier because a limited number of sites are correcting and cleaning source data, making database updates and improvements, etc. Because all of the sites use the same toolsets quality control and configuration management, work done at any site is of immediate use to all other sites. Reuse is more explicitly captured in terms of the architectural principles, wherein a master database and common Application Programmer Interface (API) provide access to the same cleaned, correlated data sets for all end users. Basically, the source data cleaning and correction phase only needs to be executed once, and then updated only as needed for reuse across all SE Core programs and applications. Finally, because database generation is centralized, it will be much easier to identify which databases and areas have been built or processed before.

## 4.4 Data-Driven, Flexible

To fully support a centralized capability providing commonality across many programs and applications, the resultant process must be flexible, data-driven, and extensible. The STDGC must be able to support a wide range of possible source data, and it cannot be oriented around specific end-application needs. Where many database generation systems have highly specialized capabilities oriented toward a particular image generator (IG) or computer generated forces (CGF) application, the STDGC must be able to handle the superset of these needs. However, when data is ultimately exported for use in run-time applications, the STDGC must be capable of presenting a view of the data that meets all of the highly-specific application needs in that data instantiation.

As new programs are supported or new requirements arise, the STDGC must be extensible to allow new functionality, tools, data types, and processing procedures. Similarly, the STDGC must be able to grow as various SE Core requirements are addressed and as new technologies are developed.

## 4.5 Correlated end products

The STDGC is required to support interoperability among SE Core confederate programs by providing mechanisms to specify what type of data can be exported for a particular training use case. For example, consider the case of two confederate programs that use different IGs with different polygon density budgets. When one of those programs is to be used for training by itself, it will specify its system capability to the STDGC and get back run-time databases that match those density, feature type, and polygon budget rules. However, for a training use case wherein both of the programs must interoperate, their differing requirements must be weighed against each other to provide tradeoffs for interoperability. Thus, for the interoperable case, each program's run-time databases would be exported with somewhat different content than they would see in a standalone use case. As a simple example, the program with the lowest polygon budget might be used to limit the program with the higher polygon capacity, thus resulting in the same lower density "view" of the simulated environment for each application.

This is possible because the STDGC will provide a single process for the generation of a master database that is then used to provide data to various programs. Because a common data-driven process exists, basic source-driven correlation errors between end database products can be mitigated. It is recognized that this capability is not always a simplistic tradeoff of "least common denominator". The "least common denominator" is defined as the greatest feature set that all target systems are capable of processing with out severely impacting performance. For example, alternate levels of abstraction can be used in certain cases, such as a tank simulator operating on data that does not represent the details of a building interior (perhaps limiting its view to information on external openings), while an individual combatant simulation stores and operates on the building interiors. In this case, the entire functional purpose of the individual combatant simulator might be nullified if a straightforward least-common-denominator approach is used (i.e. no building interiors at all).

While there are challenges to address, a successful implementation of this concept will eliminate one of the fundamental causes of miscorrelation between simulations. SE Core's overarching goal of interoperability and fair fight among confederate programs will be supported by this capability, since correlation of terrain databases ("same place") is a key precursor for interoperability.

## 4.6 Timely database generation

The STDGC will be built from the ground up to support automated, rapid database generation while simultaneously recognizing the need for data improvement tasks that require longer production timelines and human intervention. A largely-automated capability can be used to run from source data to run-time databases very quickly, with follow-up phases wherein SE Core personnel incrementally improve the datasets with manual effort.

### *4.6.1 Rapid Database Generation*

A key principle of the STDGC concept is rapid turnaround of databases from available source to run-time formats. The threshold requirement is expressed in terms of 96 hours, while an objective is cited at 72 hours. These timelines are initiated from the availability of valid (processed) vector source data from a source data provider such as NGA through to creation of selected output formats (e.g. run-time databases).

In cases where run-time databases are required for an area already covered by the STDGC master database, then processing is limited to the creation of the appropriate run-time databases based upon identification of parameters for database content. However, if generation is required from source data, the master database must be extended via upfront processing to provide the necessary content. This process will be automated to the extent possible.

The principle of rapid database generation has been represented thus far as a time and area/density (e.g. "96 hours for 180km square, Digital Terrain Elevation Data (DTED) Level 3"). However, rapid generation is inevitably a matter of tradeoffs beyond area and density, including quality, content, usability, correlation with the real world, etc. Thus, while the STDGC will have the capability to process from raw source data to run-time formats in a compressed timeline, it is understood that this will have implications such as data quality limitations. The STDGC must thus support automation to the extent possible, while allowing means to incrementally update data as time allows for manual corrections or incorporation of improved source data.

#### *4.6.2 High Quality, Manual Production*

The STDGC will include organic capabilities to repair source data errors, reconcile conflicting source, generate visual models, create geo-specific datasets, create building interiors with little or no source available, attribution of the source data to comply with confederate EDM, etc. While automation opportunities will be pursued for any and all of these through spiral development and technology advances, it is recognized that manual effort will be required for the near term. Similarly, input and review of data by Subject Matter Experts (SMEs) or end users may be required in some cases. As a result, the STDGC will have a robust capability for operator intervention. At any time, it will be possible to halt manual effort and switch to a rapid production mode, wherein automated processes take over. This provides the STDGC with the ability to be responsive to short timelines, while taking full advantage of any time available to make incremental improvements.

### 4.7 Cost effective

A corollary of all of the above principles is cost effectiveness. If all PEO-STRI simulations can productively make use of the same, centralized facilities to turn around databases in just a few days, then clearly infrastructure, training, development, and even direct labor costs will be minimal to the end user.

As with any change, there will be an up-front cost for each confederate program (or application) to integrate into the SE Core architecture. This is a two-way relationship, with both sides incurring a cost and a benefit. SE Core must adapt to meet new or specific program needs, but gains another confederate program with which all other confederate programs can interoperate. The confederate program gains reuse and interoperability from SE Core, but must integrate a common architecture. Specifically in the area of database generation, SE Core will take on the burden of creating new databases (from source to run-time formats). New confederate programs must provide detailed data requirements and integrate datasets produced by SE Core. In some cases, confederate programs may need to eliminate or work-around highly specific data requirements. As a critical mass of capabilities and programs are integrated into SE Core, additional programs will be less likely to have specialized requirements. Some new programs may discover that many of their run-time formats are already supported by SE Core, thus incurring reduced costs in return for access to all of the SE Core databases

### 4.8 Production-Ready Technologies

SE Core is not a research platform, but rather a practical solution to support training. As a result, the SE Core database generation capability must be based upon mature technologies, taking advantage of proven COTS solutions where possible. It is recognized, however, that to achieve all SE Core database generation goals some development will be necessary. As a result, development must be phased to ensure solid, production-ready capabilities at each incremental step. This solid foundation of capabilities will be incrementally developed over time. To support long-term growth, the process must be modular and flexible, such that technology insertion is possible as capabilities mature.

### 4.9 Encapsulation

Because the SE Core database generation capability is maintained and executed at central sites, it is not necessary for the internal components of the process to be accessible to outside parties.

This provides flexibility within the database generation capability, allowing internal processes, formats, and tools to change with minimal implications to data producers or confederates. This is achieved through the use of standard inputs and a plug-in architecture for exports of various formats. The source data used as input to the Database Generation Capability is based upon widely-used, standard formats. A wide range of tools and processes are available to operate on this data within the SE Core process, providing flexibility. A plug-in architecture is provided so that plug-ins can be developed to accept SE Core data and organize it for export to various end-use formats. SE Core's internal toolsets, formats, and processes are wholly hidden from data producers and export plug-ins, thereby allowing for internal changes as needed.

### 4.10 Automated Test and Description

The SE Core database generation capability will include extensive support for rapid, automated, in-line testing and data mining. This will include the capability to characterize data, such as feature counts, automatic identification of dense regions, quick access to features by type or attributes, etc. Similarly, automated test capabilities will help characterize the overall quality and complexity of the dataset. Finally test capabilities will take full advantage of machine readable requirements, such as Environment Data Models, to verify data content against programmatic needs.

## 5. STDGC Key Architectural Concepts

Portions of the above key principles have been translated into architectural concepts. Generally speaking, these concepts have been used to articulate SE Core goals in a more concrete form.

The STDGC architecture is envisioned as being composed of five key pieces as described in the following sections (see figure 4). Two of these sections represent "external interfaces" from SE Core's perspective, with responsibility shared by SE Core and multiple outside parties. The first external interface (input) is the Common Dataset, which includes source data such as that provided by NGA. The second external interface (output) is with confederate programs or applications. While the run-time databases themselves are fully in the domain of programs and vendors, plug-ins must be provided to create those run-time databases at SE Core sites. Thus, the core components of the STDGC itself (wholly within SE Core's responsibility) are a Master Terrain Database Generation Toolset, the Master Database, and a plug-in architecture. These concepts are illustrated in Figure 4.

Although it is not represented as a separate entry in this section, the concept of multiple database generation facilities (one central site supplemented by satellite sites) is key to the overall SE Core concept. While the following architectural components could be implemented in many ways, the view of centralization provides critical context. For example, if the STDGC vision is fully realized, then the concept of data "interchange" would become largely meaningless for confederate simulation programs: all data would come from one of the specialist-staff central sites, all of which have the same toolsets and capabilities. "Interchange" thus evolves into external data being imported into the STDGC process….none of SE Core's confederate programs would need to know anything about how this came about

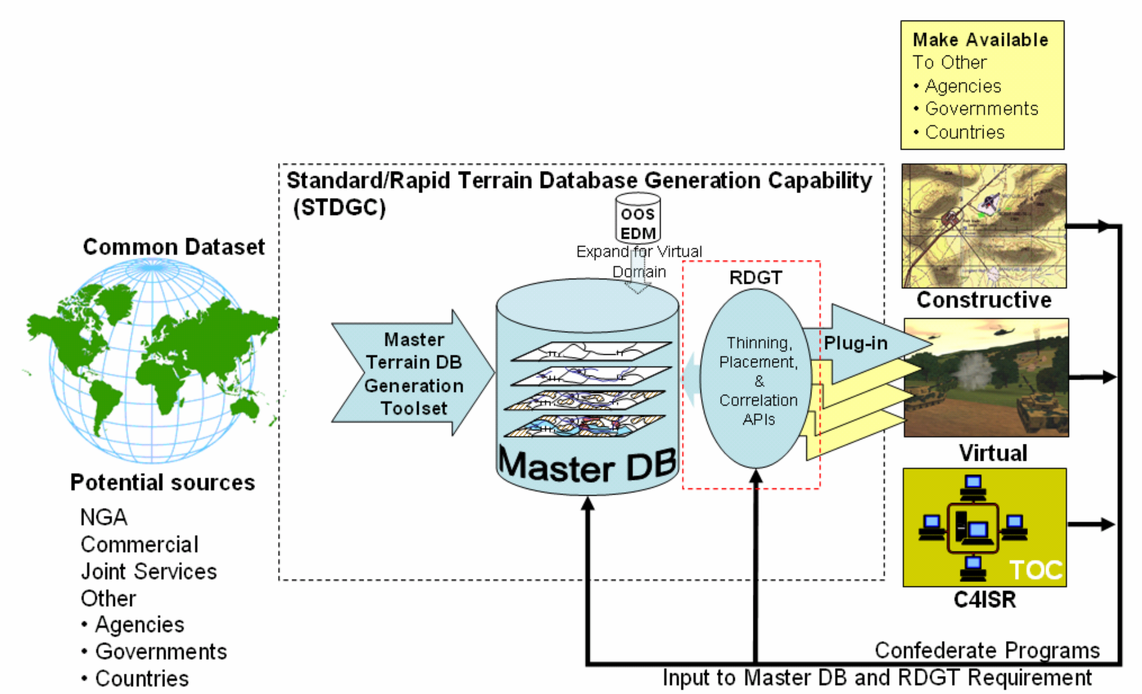


Figure 4: Architectural Components

## 5.1 Common Dataset (Source)

The SE Core program is assuming that timely, high quality, properly attributed source data will be available for use by the STDGC. Data providers could include NGA, Commercial, and Joint Services. The STDGC will support the current set of common industry formats and government source formats, while also including a flexible architecture that will support future formats as needed. SE Core itself is not responsible for creation of the common dataset (i.e. source). It is expected that Memorandum of Understanding (MOUs) will be generated between the SE Core maintainers and source data owners to understand data distribution restrictions, data requirements (e.g. attribution), and the requirements for data quality.

The "Common Dataset" is thus all sources that are supported by STDGC and which provide sufficient content and quality to meet STDGC needs. Management of the common dataset, which includes quality assurance (QA) procedures, configuration management, and distribution control, is an SE Core function. Thus, the STDGC must include tools to organize, present, configuration manage, and control access to sources.

## 5.2 Master Terrain Database Generation Toolset

The Master Terrain Database Generation Toolset (MTDGT) is responsible for converting source data from the Common Dataset into the Master Database. The primary functionality of the MTDGT is reconciling of various source data formats and ensuring the superset of data requirements for target applications is met. Thus, the MTDGT must be compliant with the union of all data needs for SE Core.

In order to meet the strict timelines established as SE Core requirements, the MTDGT must have automated capabilities to address basic content and correlation errors in available sources. For example, missing attribution must be added and sources that disagree on feature location or content must be reconciled in some way. The MTDGT does *not* specialize data (e.g. it would not “pick” between representing a forest as individual trees, a canopy, or a draped texture) instead it ensures that the necessary foundation data is present to support any required representation in a correlated manner. Similarly, the MTDGT does not thin data to meet a particular program’s density or polygon budgets; instead, it captures the highest level of detail available in source. To meet data quality and content requirements, the MTDGT may be required to eliminate data that cannot be automatically correlated between sources, and may even be required to intensify available data. Examples of MTDGT functionality would be alignment of rivers from one source with high-resolution terrain skin (or bridges) from another.

Because the Master Database is not itself a specialized run-time format, the MTDGT is not analogous to program-based database generation systems that go from source to run-time formats. Instead, it is conceptually closer to Geographical Information System (GIS) tools that organize and structure source data for further processing. Also, the MTDGT must be open and well documented so that multiple vendors can support MTDGT maintenance and technology insertion.

The MTDGT’s ultimate output is the Master Database (“ready to process”) as described in the following section.

## 5.3 Master Database

The Master Database (MDB) contains data that has been processed from source into a representation ready for conversion into run-time formats. It is not by definition a single format or a single data file; instead, it captures the concept that the STDGC itself includes processed (cleaned, refined) data that is available for reuse and export to multiple end users. The MDB concept is based upon the following:

1) Union of Requirements: The MDB contains sufficient information to provide data as required for any program that develops the plug-ins and provide them to create those run-time databases at SE Core sites.
2) Full Capture of Source Density and Complexity: The MDB contains the most comprehensive view of data available from source, even if no existing target program can handle the density or content. This improves the chance that content will be available for future needs.
3) Reuse: Once data is processed into the MDB, many end-users, applications, etc. can reuse it. After the initial database creation, the up front work to acquire and clean source data is eliminated for future applications that overlap the available area and content. However, as new or improved source data comes available, it will be necessary to update the MDB on demand.
4) Maintained by STDGC Sites: The STDGC sites would maintain the MDB, exchanging updates or additions. Updates made at one STDGC site will be verified via automated testing for quality control, placed under configuration management, then propagated to all sites for reuse.
5) Not an Exchange Format: The MDB is *not* an exchange format. It is a capture of all data necessary to provide required content through the plug-in architecture. The STDGC sites will own and maintain the processes for creating MDB content as well as the applications required to export into confederate program formats. New formats must be sponsored by

confederate programs. However, since SE Core is expecting to provide a SEDRIS export capability, this could be used to support exchange with programs that are not confederates.

6) Common Projection: The MDB will have a common spatial reference model. The MDB will have as its primary spatial reference model geodetic coordinate system and use the WGS 84 datum. Point features will be represented in a local coordinate system.

## 5.4 Run-time Database Generation Toolset

The run-time database generation toolset (RDGT) is the linchpin STDGC capability, supporting the following goals:

1) Shelter End Users: The MDB contains the best possible density and widest support for possible end-user representations. This complexity is hidden from developers of the various exporters.
2) Common Tailoring: The RDGT provides the ability to specify scenario- and system-specific requirements and have all presented data meet those requirements. Thus, different use cases or different system requirements will result in different database content being generated from the same Master Database. If a particular system can handle only a limited number of individual trees (preferring canopies to represent forests), then individual trees beyond a certain density would not be presented for export to that system's databases. Instead a correlated canopy representation would be provided. Similarly, IG-specific requirements, such as material encoding for sensors or polygon densities, would be explicitly supported through the RDGT. The Master Database's "high detail" view of the data would be thinned and reduced as necessary to meet program-specific requirements.
3) Support for Interoperability: Because the RDGT can be tailored for content, a set of programs which must interoperate with each other can generate specialized run-time databases all built with, for example, the same "least common denominator" requirements. This would be accomplished by using exporter plug-ins to build their unique run-time databases, but using the same parameters telling the RDGT to provide a common data view. Thus, the data stored in each system would be different than they might see for the use case where they are running by themselves. This capability does not ensure interoperability, but is a stepping-stone toward controlling correlation errors.
4) Common Spatial Reference Model: The RDGT must support a common geospatial transformation method to transform spatial data to the required target's spatial reference frame definition. It is suggested that the SEDRIS Spatial Reference Model (SRM) be used to accommodate this. (See: http://sedris.org/srm.htm )
5) The RDGT must be documented so that additional developers can add to, maintain and insert new technology into the RDGT. To this end, the interface to the MDB, the plug-in interface and the methodology for writing rule based data manipulation methods must be documented to the extent that would allow third parties to add new software to RDGT.

To meet these goals, the RDGT is composed of four capabilities. First, it provides a mechanism for SE Core database personnel to specify the data requirements for a particular export. Second, it imports the master database for processing. Third, it "specializes" the data based upon user-specific requirements. Finally, it provides a plug-in architecture which pushes data to vendor-supplied export plug-ins. Each of these four key subcomponents are described below:

### *5.4.1 Specify Requirements*

The STDGC as a whole will require a robust requirement capture mechanism, focused on machine-readable requirements for database content, program-specific data structures and rules, etc. These requirements will then drive the content generated for storage in the Master Database. Because the Master Database incorporates a superset of capabilities, no single exporting application will want to be exposed to the full scope of data. Thus, it is necessary to specify the "use case" requirements, which are a mixture of basic requirements (which tend to stay the same for a particular format or confederate program) as well as requirements unique to a specific export of data (like database extents, or providing a different default for missing attributes such as river depth). Similarly, in cases where multiple confederate programs must interoperate, it will be necessary to specify tradeoffs between different confederate capabilities, representations, and content. This concept is discussed in more detail in section 4.5.

Export requirements may be specified with a wizard or GUI, as this concept is illustrated in Figure 5: RDGT Concept.

To summarize, the end result of STDGC-level requirements is to ensure that all required supporting data is available in the Master Database. The end result of requirements specified at the RDGT level is to "specialize" the data that is exported for a particular use case.

### *5.4.2 MDB Read*

The Master Database Read box (API) is simply intended to illustrate that the RDGT must be able to read in the master database content.

### *5.4.3 Database Specialization*

The RDGT's key component is the Database Specialization capability. The database specialization capability operates on the data loaded from the Master Database based upon the use case and programmatic requirements provided to push data to the various exporter plug-ins. The processing step must be capable of thinning, intensifying, and reorganizing data based upon the export requirements. It must also be able to derive representations as required for the target exporters: trees versus canopies, TINs versus grids, building shells versus building interiors, areal rivers versus linear rivers, etc. Similarly, it must provide, perhaps via defaults, any required attribution that is not present in the master database. Any alteration of data between MDB read and "push" to the exporter plug-ins must be deterministic: given the same input data and the same parameters, the same data is pushed to the exporter plug-ins.

Where the MTDGT handles the "union" or "superset" requirements, the RDGT is used to push "intersection" or "least common denominator" data based upon specified parameters.

This critical capability must be developed in a balanced fashion, providing a cohesive capability between MDB generation, Master Database content, and run-time database generation (RDGT). For example, deconfliction of source errors (river flowing up hill) should be decisively addressed in the MTDGT and not in the RDGT. Similarly, if provided source contains only "city areals" and no individual buildings, but some export plug-ins require individual buildings, then it is necessary for the MTDGT to export sufficient information into the MDB for the RDGT to "derive" individual locations in a repeatable (deterministic) fashion. For example, the MTDGT could export boundary, type (residential, commercial), and density into the MDB. From this, the MTDGT must be able to derive realistic patterns of building placement, perhaps based upon a user-specific seed.

### 5.4.4 ***Data Push Through Plug-In Architecture***

Once the database specialization capability has created appropriate content, it is pushed to the various export plug-ins (described in the next section). Here, a public plug-in architecture (Synthetic Environment Extraction (SEE Plug-in API)) is specified in detail to support developers of plug-ins. Data is organized and structured for presentation to the plug-ins. The plug-ins then format the data as required for their target format.

A notional graphical illustration of RDGT concepts is captured in Figure 5. Here, a GUI is shown capturing a few of the possible input parameters that might be specified when determining the "view" of the data to be provided. Different parameters would result in different data being provided, even for the same geographic area.

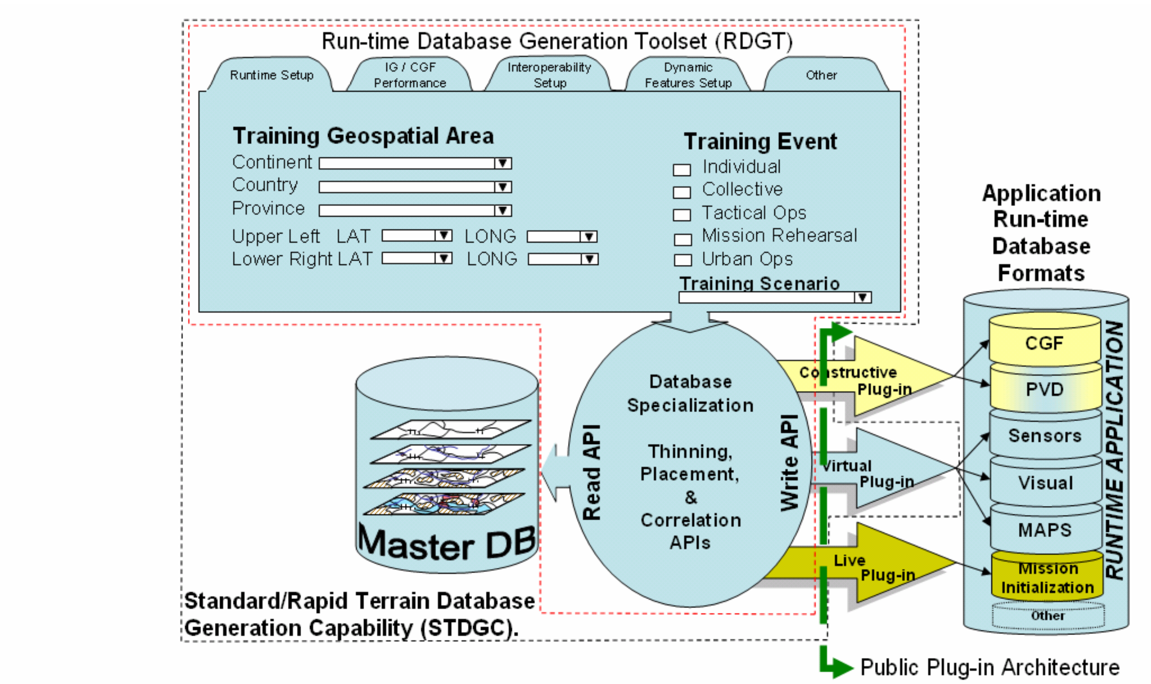


Figure 5: RDGT Concept

## 5.5 Exporter Plug-ins and Run-time Databases

Run-time database creation is not exclusively the responsibility of the SE Core. It is incumbent upon SE Core to provide all necessary access and data through a public plug-in architecture, while individual data consumers must provide export plug-ins for their various formats, even if proprietary or program-specific. These plugins will have government purpose rights. As an export plug-in is developed, it will be the joint responsibility of SE Core and the plug-in developers to identify and resolve issues. Once the plug-in is completed, it will be available at the various SE Core MDB Generation Centers for future use. SE Core personnel will be responsible for executing the database generation capability to export various run-time formats.

Thus, output from the STDGC is databases (i.e. the output of export plug-ins), not access to the SE Core data.

In this manner, the STDGC will grow an organic capability to export a wide range of target formats. Supported formats will grow as new confederate programs are identified and incorporated.

## 6. STDGC Dependencies and Assumptions

The SE Core database generation capability is predicated on several key assumptions:

1) Source data must be provided with adequate attribution, content, and quality to allow largely automated processing of databases on a short timeline.
2) Validation of source data is the responsibility of source data providers.
3) MOUs must be established with source data providers to document above assumptions, along with data distribution rights and availability.
4) To meet the need for the 96 hour process, the end users (simulations) must complete validation of their own runtime databases for the mission at hand.
5) Close coordination and dedicated support from confederate systems will be required. For specialized database formats, they must develop and maintain formatters and/or format specifications to facilitate development in support of evolving STDGC capabilities and evolving functionality within the confederate systems.
6) Security concerns will have to be addressed in collaboration with confederate programs. Databases must be built in a secure environment, including the ability to generate classified and non-classified data.
7) An ACA will be required between the STDGC contractor(s) and each confederate program vendor.

## Appendix A: Abbreviation/Acronym Dictionary

| Acronym/<br><br>Abbreviation | Description |
|---|---|
| API | Application Programmer Interface |
| AVCATT | Aviation Combined Arms Tactical Trainer |
| C4ISR | Command, Control, Communications, Computers, Intelligence Surveillance and Reconnaissance |
| CATT | Combined Arms Tactical Trainer |
| CBNRE | Chemical Biological Nuclear Radiological Effects |
| CCTT | Close Combat Tactical Trainer |
| CGF | Computer Generated Forces |
| COTS | Commercial Off The Shelf |
| CVE | Common Virtual Environment |
| DTED | Digital Terrain Elevation Data |
| FCS | Future Combat System |
| GFI | Government Furnished Information |
| GIS | Geographical Information System |
| IA | Integrated Architecture |
| ICD | Interface Control Diagram |
| IG | Image Generator |
| JIIM | Joint, Interagency, Intergovernmental, and Multinational |
| JSAF | Joint Semi-Automated Forces |
| LVC | Live/Virtual/Constructive |
| MDB | Master Database |
| MTDGT | Master Terrain Database Generation Toolset |
| MOUs | Memorandum of Understanding |
| NGA | National Geospatial Agency |
| NRFTT | Network Reconfigurable Full Task Trainer |
| OneSAF | One Semi-Automated Forces |
| OOS | Objective OneSAF System |
| OTB | OneSAF Testbed Baseline |
| PEO STRI | US Army Program Executive Office Simulation, Training, and Instrumentation Command |
| PTT | Part Task Trainer |
| QA | Quality Assurance |
| RDGT | Run-time Database Generation Toolset |
| SAIC | Science Applications International Corporation |
| SE | Synthetic Environment |
| SEE | Synthetic Environment Extraction |
| STDGC | Standard Database Generation Capability |
| STF | SEDRIS Transmittal Format |
| TDB | Terrain Database |
| TE | Training Environment |
| WAN | Wide Area Network |
| WARSIM | War Simulation |